



Attorney Docket No. 30014200-1027

PATENT

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re Application of:	)	
	)	
Hideki Hiura et al.	)	Group Art Unit: 2126
	)	
Application No.: 09/488,909	)	Examiner: Phuong N. Hoang
	)	
Filed: January 21, 2000	)	
	)	
For: METHOD FOR ENABLING	)	
MULTIPLE CONCURRENT	)	
SUBPROCESS HANDLING	)	
ON A SYSTEM USING A	)	
GLOBAL PROCESS	)	

Mail Stop AF  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**PRE-APPEAL BRIEF REQUEST FOR REVIEW**

Sir:

In accordance with the Pre-Appeal Brief Conference Pilot Program introduced in the Official Gazette (July 12, 2005), Applicants respectfully request a pre-appeal brief review. Applicants request a review of the final rejection in the above-identified application. No amendments are being filed with this request. A Notice of Appeal is submitted herewith.

**I. STATUS OF CLAIMS**

Claims 1-21 are pending in the above-identified application. Claims 1-21 were rejected in the Final Office Action dated April 19, 2005.

**II. STATUS OF AMENDMENTS**

A Request For Reconsideration After Final was filed on August 19, 2005, wherein no amendments of the claims were made. An Advisory Action was mailed September 13, 2005, stating that the Request For Reconsideration After Final had been considered but rejected because the Request did not place the application in condition for allowance.

**III. ISSUES**

The issues to be reviewed are whether claims 1-11 and 15-21 are properly rejected under 35 U.S.C. § 103(a) as unpatentable over *Hetherington et al.* (U.S. Patent No. 6,275,810, hereinafter “*Hetherington*”) in view of *Kaufman* (U.S. Patent No. 5,313,647), and whether claims 12-14 are properly rejected under 35 U.S.C. §103(a) as being unpatentable over *Kaufman* in view of *Hetherington*.

**IV. ARGUMENT**

Applicants respectfully submit that the Examiner has omitted one or more essential elements needed for a *prima facie* rejection. The cited art, alone or in combination, fails to teach or suggest every limitation of the claims. For example, the combination of *Hetherington* and *Kaufman* fails to teach or suggest at least “mapping a user-specific process so that it overlays virtual addresses of the master process,” as recited in claim 1.

In the Amendment filed November 24, 2004, Applicants argued that the combination of *Hetherington* and *Kaufman* fails to teach or suggest the limitations of claim 1 because, contrary

to the assertions of the Examiner, *Kaufman* fails to teach or suggest that a child process overlays the virtual address of a parent process. In response to Applicants' arguments that the duplication of information, or "forking," disclosed in *Kaufman* is not the same as overlaying the same virtual address, the Examiner asserted that "Kaufman teaches the mapping so that the child overlaid the virtual address of the parent (vm\_fork, syscall\_finish\_fork, vm\_mapin to mapin [sic] the overlay object and mapping out a file range from a process's context address space, col. 31 lines 15-65 and col. 34 lines 10-15)" (Final Official Action of 4/19/2005, p. 8, no. 24). However, the Examiner's argument with regard to mapping in and out of a process's context address space still fails to establish *prima facie* obviousness.

The portion of *Kaufman* cited by the Examiner states that "[t]he VM system executes the steps of MAPOUT as a user entry for mapping out a file range from a process's context address space. (See Col. 34, ll. 10-15 of *Kaufman*). The procedure accepts as input a handle of the map, and returns a status of the mapping procedure" (Emphasis added). However, claim 1 recites a virtual address overlay and not a context address overlay or mapping. A context address is very different from a virtual address, as explained in *Kaufman* itself:

The memory architecture of system 10 consists of two levels of related address space: context address (CA) space and system virtual address (SVA) space. Context address space is the programmer's interface to memory. There are many context address spaces in a system. System virtual address space stores the data from all context address spaces. There is only one system virtual address space. Another address space, the system physical address space (SPA) defines hardware control registers. (Col. 18, ll. 29-38)

Thus, the MAPIN and MAPOUT features of *Kaufman* merely allow forked processes to map to a section of context address space of a parent process, and do not provide mapping to the virtual address of a parent process. (See Col. 30, ll. 4-14 and Col. 33, ll. 25-30 of *Kaufman*).

In the Advisory Action mailed September 13, 2005, the Examiner contended that the virtual address space stores all data from the context address space and that each segment of the context address space is mapped to the system virtual address space. The Examiner further asserted that when a process's context address space is mapped, the process's virtual address space is also mapped. That assertion is false for at least two reasons. Firstly, *Kaufman* does not at all teach or suggest that the virtual address space is mapped out with the context address space. *Kaufman* merely discloses that the VM system executes the steps of MAPOUT as a user entry for mapping out a file range from a process's context address space (*See* Col. 34, ll. 10-15 of *Kaufman*). Secondly, there is no per-process virtual address space in *Kaufman*. *Kaufman* explicitly states that there is only one virtual address space. (*See* Col. 18, ll. 35-36 of *Kaufman*).

Furthermore, *Kaufman* discloses that "[t]he vm system executes the steps of the procedure vm\_fork to duplicate a parent process's virtual memory information for a child process." (*See* Col. 31, ll. 14-21 of *Kaufman*). However, the duplication of information is not the same as sharing or overlaying the same virtual addresses.

As a result, Applicants submit that claim 1 is patentable for at least the above-mentioned reasons. Claims 2-7 depend on claim 1 and are therefore patentable at least for the same reasons. Claim 15 is patentable for at least the same reasons as claim 1. Furthermore, claims 16-21 depend on claim 15 and are therefore patentable at least for the same reasons.

Applicants further submit that claims 8 and 12 are also patentable for at least the same reasons as claim 1. Just as *Kaufman* fails to teach or suggest "mapping a user-specific process so that it overlays virtual addresses of the master process," *Kaufman* also fails to teach or suggest mapping "a user-specific process to virtual addresses that mirror virtual addresses of the global

process.” Furthermore, claims 9-11 and 13-14 depend on claims 8 and 12 respectively and are therefore patentable at least for the same reasons.

## **II. Conclusion**

In view of the above remarks, Applicants submit that all claims are allowable over the cited prior art and respectfully request early and favorable notification to that effect.

Respectfully submitted,



---

A. Wesley Ferreebee  
Registration No. 51,312

Dated: October 18, 2005

Customer No. 26263  
SONNENSCHN NATH & ROSENTHAL LLP  
P.O. Box 061080  
Wacker Drive Station  
Chicago, Illinois 60606-1080  
Telephone: 202/408-6832  
Facsimile: 312/408-6399